

Taller de Programación en Python: Sistema de Gestión de Biblioteca

Objetivo del Taller

Desarrollar un sistema básico de gestión de una biblioteca que permita a los usuarios gestionar libros y realizar acciones básicas como agregar, buscar y listar libros. El sistema deberá implementarse utilizando programación orientada a objetos en Python.

Requisitos del Taller

1. Clase Libro:

- Atributos:
 - título
 - código
 - autor
 - año_publicación
 - disponible
- Métodos:
 - disponibilidad: Método que devuelve un mensaje sobre la disponibilidad del libro.

2. Clase Biblioteca:

- Atributos:
 - libros (lista de objetos Libro)
- Métodos:
 - agregar_libro: Método que permite agregar un objeto Libro a la lista libros.
 - buscar_libro: Método que recibe título o código y devuelve el libro correspondiente si existe.
 - listar_libros: Método que lista todos los libros en la biblioteca y su disponibilidad.
 - prestar_libro: Método que permite prestar un libro, debe verificar que el libro esté disponible.
 - devolver_libro: Método que permite devolver un libro, cambiando su estado a disponible.

UNIVERSIDAD MARIANA
FACULTAD DE INGENIERÍA
PROGRAMA INGENIERÍA DE SISTEMAS
CURSO: Python II

Desarrollo del Taller

1. Crear la clase Libro con los atributos y métodos especificados.
2. Crear la clase Biblioteca con los atributos y métodos especificados.
3. Instanciar objetos de la clase Biblioteca.
4. Implementar un menú interactivo que permita al usuario elegir entre las siguientes opciones:
 - Agregar un libro
 - Buscar un libro por título
 - Listar todos los libros
 - Prestar un libro
 - Devolver un libro
 - Salir del programa
5. Uso de condicionales y ciclos:
 - Usar condicionales para validar las opciones seleccionadas por el usuario.
 - Implementar ciclos para mantener el menú activo hasta que el usuario decida salir.
6. Funciones:
 - Definir las funciones necesarias dentro de las clases para realizar las acciones requeridas.

Entregables

- **Código fuente en Python (.py)**
- **Documentación breve que explique cómo usar el sistema y las decisiones de diseño tomadas.**

Evaluación

- **Funcionalidad:** El programa cumple con todos los requisitos establecidos. (2.0)
- **Uso de Estructuras:** Se utilizan correctamente condicionales, ciclos, funciones, POO(1.0)
- **Claridad del Código:** El código es legible y está debidamente comentado.(1.0)
- **Validaciones:** Se requiere que se realicen las validaciones pertinentes en cada caso (1.0)